

***Amendments to the Claims***

1. (Currently Amended) A method for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising the steps of:

defining the geometry chunk with a bounding region when said bounding region has not already been defined for the geometry chunk in a previous frame, wherein said bounding region defines ~~the~~ a space the geometry chunk occupies on the compositing window;

sending said bounding region to ~~each of the subareas~~ graphics pipelines to determine ~~which subarea~~ a graphics pipeline of said graphics pipelines that will render the geometry chunk defined by said bounding region, wherein a subarea in the compositing window is assigned to receive an output of said graphics pipeline; and

communicating the geometry chunk to ~~the subarea~~ said graphics pipeline that will render the geometry chunk.

2. (Currently Amended) The method of claim 1, wherein the geometry chunk is comprised of ~~one or more pieces~~ a piece of a geometry provided by a graphics application.

3. (Currently Amended) The method of claim 1, wherein said space is a screen space.

4. (Currently Amended) The method of claim 1, wherein said space is a world space.

5. (Currently Amended) The method of claim 1, wherein said space is an object space.
6. (Canceled)
7. (Canceled)
8. (Original) The method of claim 1, wherein the geometry chunk is represented as a display list.
9. (Original) The method of claim 1, wherein the geometry chunk is represented as a vertex array object.
10. (Original) The method of claim 1, wherein the geometry chunk is represented as buffered vertices.
11. (Currently Amended) A system for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising:
  - a geometry distributor that defines a bounding region for the geometry chunk when said bounding region has not already been defined for the geometry chunk in a previous frame, wherein said bounding region defines ~~the~~ a space the geometry chunk occupies on the compositing window; and

~~one or more graphics units pipelines, wherein said graphics units are assigned to the subareas in the compositing window,~~ wherein said geometry distributor sends said bounding region to ~~each of said one or more graphics units pipelines~~ to determine which a graphics unit pipeline of said graphics pipelines that will render the geometry chunk defined by said bounding ~~region; region, wherein a subarea in the compositing window is assigned to receive an output of said graphics pipeline,~~ and wherein said geometry distributor communicates the geometry chunk to said graphics ~~unit pipeline~~ that will render the geometry chunk.

12. (Original) The system of claim 11, further comprising a graphics application that provides the geometry chunk to said geometry distributor.

13. (Original) The system of claim 12, wherein said geometry distributor comprises a virtual graphics unit that interfaces with said graphics application.

14. (Currently Amended) The system of claim 11, wherein the geometry chunk is comprised of ~~one or more pieces~~ a piece of a geometry provided by a graphics application.

15. (Currently Amended) The system of claim 11, wherein said space is a screen space.

16. (Currently Amended) The system of claim 11, wherein said space is a world space.

17. (Currently Amended) The system of claim 11, wherein said space is an object space.

18. (Currently Amended) The system of claim 11, wherein said geometry distributor comprises:

a bounding region calculator that calculates said bounding region for the geometry chunk;

a graphics ~~unit~~ pipeline assignor that assigns said graphics ~~units~~ pipeline to the ~~subareas~~ said subarea in the compositing window; and

a graphics ~~unit~~ pipeline distributor that distributes the geometry chunk to the ~~appropriate graphics unit~~ pipeline.

19. (Original) The system of claim 11, wherein the geometry chunk is represented as a display list.

20. (Original) The system of claim 11, wherein the geometry chunk is represented as a vertex array object.

21. (Original) The system of claim 11, wherein the geometry chunk is represented as buffered vertices.